Chapter 3.2
Prediction of Non-Functional Properties of Service-Based Systems:
A Software Reliability Model

Adel Taweel
King's College London, UK

Gareth Tyson
King's College London, UK

ABSTRACT

From the outset, service-based systems offer several advantages, including the promise of shortening the development life cycle, reducing the costs of software development and faster utilisation of recent technical improvements in the software industry in terms of capability, reliability, compatibility, performance, and so forth. However, this is rarely realised in practice. An important issue of service-based systems is that they are likely to be (or perhaps already being) used in domains where human life and/or economic loss are possible and the need for a highly reliable system is a must. However, would it be possible to build reliable service-based systems that meet such domains' requirements? Does it necessarily mean that composing a system from highly reliable services produce a highly reliable system? Is it possible to predict quality attributes of a service-based system from its services before building it? In this chapter, the complexity of this issue is highlighted, focusing on reliability, in an attempt to answer some of these questions. The chapter outlines various approaches that attempt to address this issue, and proposes a possible way forward for predicting the reliability of service-based systems from its individual services.

DOI: 10.4018/978-1-61350-456-7.ch3.2

INTRODUCTION

Traditionally, software systems are constructed from individual parts, which are then integrated to produce a system that meets certain functional and non-functional specifications. The produced system, as a whole, is then tested to check whether these specifications are met. Systems in this case are therefore normally viewed as a whole and thus tagged by these specifications, rather than as individual parts. Although changes may need to be done on certain parts of the system to improve a non-functional property, the whole system will have to be re-tested to check whether or not that improvement has been achieved.

In service-oriented architectures, however, systems are constructed or integrated from individual services. These individual services are combined together, normally based on a defined architecture or framework, to produce a system that meets certain functional and non-functional specifications or properties. These individual services, on the other hand, have their own functional profile, run-time environment, and functional and non-functional properties. Which leads to the question: *whether the properties of individual services can be used or extended to determine the system-level properties of the composed system?*

One way to determine the properties of the composed system is to re-test this composed system as whole. However, in service-based systems, this method is not always feasible and is not preferable for several reasons. Firstly, with services that are expected to be bought (or obtained) individually, possibly from different vendors, it is unlikely that developers (or integrators[1]) would commit themselves to buying a service that they might later discover to not achieve the specific purpose that it was originally bought for. Secondly, even if it were possible to acquire a service (or a use of a service) without buying it, perhaps through a provided evaluation period, it is less likely that integrators would be happy to waste their time and effort on incorrectly selected services. Thirdly, the longer the service selection process takes, the less economical it becomes for integrators as they are expected to meet deadlines and budget.

In reality, integrators are unlikely to choose a service to test or evaluate for their system unless it stands a fair chance of success. The selection of a service can therefore be a complex task, and a mechanism that facilitates the selection, even to a certain degree, would greatly reduce the complexity of the process and thus the development of the system. One mechanism could be envisaged from enabling developers (or integrators) to determine or predict the system-level properties of the resultant composed system from the properties of its individual services. The potential and importance of prediction theories for service-based and component-based systems has been recognised early in the software engineering research community. In fact, special teams have shown special interest in this field, such as [Bachmann et al, 2000, Crnkovic et al, 2001, Wallnau, 2003].

The chapter first presents the background, related issues and complexities of the prediction of non-functional properties. It then presents related work outlining general approaches and potential approaches to address these issues. The rest of the chapter proposes a model to predict the software reliability of service-based systems from their individual services as a potential method to address other non-functional properties.

BACKGROUND

Large scale distributed service oriented systems are increasingly becoming used in industry and commercial settings. They are offering the rapid development throughput that software industry needs to stay competitive. However, one of the main issues is creating re-usable services that can be used to serve different purposes and different types of systems in different domains. Standardisation is a key issue for providing standard interfaces to enable flexible integration [Wallnau, 2003]. As different